

Análisis de Performance y Consumo sobre Sistemas Embebidos Multinúcleo

Dell'Oso Matías¹, Medina Santiago¹, Romero Fernando¹, De Giusti Armando^{1,2}.

¹Instituto de Investigación en Informática LIDI (III-LIDI), Facultad de Informática, Universidad Nacional de La Plata, 50 y 120 2do piso, La Plata, Argentina.

²CONICET – Consejo Nacional de Investigaciones Científicas y Técnicas
{mdelloso, smedina, fromero, degiusti}@lidi.info.unlp.edu.ar

Resumen. El avance tecnológico de los últimos años y la cantidad de datos que actualmente un sistema debe procesar, les exigen a los sistemas de tiempo real actuales una mayor capacidad de cómputo. Para hacer frente a esta demanda, surgieron los procesadores multinúcleo orientados a aplicaciones embebidas. En el siguiente trabajo se estudiarán puntualmente los sistemas multinúcleos asimétricos, y se presentará un análisis comparativo entre dos procesadores pertenecientes a la familia Cortex-M. Los puntos de análisis incluyen los tiempos de latencia entre núcleos, utilizando distintas técnicas de comunicación y el consumo de energía.

Palabras clave: Sistemas Embebidos, Multinúcleo, Consumo de Energía, Sistemas Heterogéneos, Cortex-M, Tiempo Real, Proyecto CIAA.

1. Introducción

En su mayoría, los Sistemas de Tiempo Real [1][2][3] están basados en microcontroladores, pequeñas unidades de procesamiento dedicadas de bajo consumo y con gran cantidad de periféricos integrados para el control de sensores y actuadores.

Hoy en día, con el auge del Internet de las Cosas y el avance tecnológico, la cantidad de datos que un sistema debe procesar ha crecido exponencialmente en los últimos años [4]. Las tecnologías actuales les exigen a los sistemas en general y, especialmente a los sistemas de tiempo real, una mayor capacidad de procesamiento. Esto llevó a que, en enero de 2014, NXP¹ desarrollara el primer microcontrolador multinúcleo asimétrico basado en la familia Cortex-M [5][6][7]. El objetivo principal de este tipo de procesadores es contar con un procesador de alto rendimiento y un coprocesador con un poder de cómputo más reducido. La principal ventaja es poder reducir la carga de trabajo del núcleo principal. Por ejemplo, mientras el procesador de alto rendimiento se encuentra procesando datos, el coprocesador podría estar

¹ NXP (Next eXPerience) Semiconductors es una empresa fabricante de semiconductores que se creó en el año 2006 a partir de la división de semiconductores de la empresa Philips. En 2015, NXP adquirió la empresa estadounidense Freescale Semiconductor, uno de los mayores fabricantes de semiconductores de ese momento.

manejando la entrada y salida del sistema. Otra gran ventaja de este tipo de sistemas es el ahorro de energía, donde podría utilizarse el coprocesador para realizar los cálculos habituales y, eventualmente, despertar al procesador principal para realizar alguna acción compleja, como por ejemplo manejar la pila TCP/IP o hacer procesamiento de señales. De esta manera tendríamos un sistema de bajo consumo que, de ser requerido, puede realizar operaciones que requieran de un gran poder de cómputo.

Los enfoques a tener en cuenta en el análisis de los sistemas embebidos multinúcleo pueden ser varios: el manejo de la entrada/salida, la planificación de tareas entre núcleos, el manejo de memoria compartida, el pasaje de mensajes entre núcleos y, como principal punto, lograr la mejor relación consumo/rendimiento.

Con el objetivo de estudiar algunos de los enfoques mencionados anteriormente, en este trabajo se lleva a cabo una comparación de dos microcontroladores multinúcleo asimétricos desarrollados por NXP (ver Sección 2.1).

Este artículo está organizado de la siguiente manera: la sección 2 detalla el hardware utilizado para realizar las pruebas; en la sección 3 se hará mención al modelo de comunicación entre núcleos; la sección 4 muestra las metodologías utilizadas en los experimentos; la sección 5 explica las pruebas realizadas y los resultados obtenidos y, la sección 6, contiene las conclusiones y líneas de trabajo futuras.

2. Hardware utilizado

2.1. Microcontroladores

Con el fin de ubicar el trabajo en el marco del proyecto CIAA (Computadora Industrial Abierta Argentina [8]), las pruebas fueron realizadas sobre las placas de desarrollo EDU-CIAA-NXP [9] y picoCIAA [10] (ver Figura 2.1 y 2.2 respectivamente).



Figura 2.1: EDU-CIAA-NXP



Figura 2.2: picoCIAA

Ambas placas cuentan con microcontroladores multinúcleo asimétricos. La primera posee el procesador LPC4337, perteneciente a la familia LPC43xx, primera serie de microcontroladores multinúcleo basado en la familia Cortex-M que fue lanzada al mercado. La segunda está basada en el procesador LPC54102, perteneciente a la nueva línea de procesadores multinúcleo asimétricos desarrollada por NXP. Las

características principales de dichos microcontroladores se pueden observar en la Tabla 2.1.

Tabla 2.1: Características principales del microcontrolador LPC4337 y LPC54102

Características	LPC4337	LPC54102
Procesador de alto rendimiento	ARM Cortex-M4F	ARM Cortex-M4F
Procesador de bajo rendimiento	ARM Cortex-M0	ARM Cortex-M0+
Frecuencia de reloj	204 MHz (compartido)	100 MHz (compartido)
Soporte para instrucciones SIMD	Si	Si
Unidad de punto flotante por hardware	Si	Si
SRAM	136 kB	104 kB
Flash	1 MB	512 kB

Ambos microcontroladores tienen el mismo modelo en lo que respecta al núcleo de alto rendimiento (Cortex-M4F), pero difieren en el coprocesador. El Cortex-M0+ es el procesador más eficiente en lo referido a consumo energético de la familia Cortex [11]. Está construido sobre la base de su antecesor, el Cortex-M0, manteniendo el mismo repertorio de instrucciones y siendo compatible con las mismas herramientas de software (ver Tabla 2.2). Gracias a esto, el LPC54102 reduce considerablemente el consumo con respecto al LPC4337, convirtiéndolo en un procesador orientado a las aplicaciones donde el consumo es una variable central.

Tabla 2.2: Características principales de los Cortex-M4F, Cortex-M0 y Cortex-M0+

Características	Cortex-M4F	Cortex-M0	Cortex-M0+
Arquitectura	ARMv7E-M – Harvard	ARMv6-M – Von Neumann	ARMv6-M – Von Neumann
Repertorio de instrucciones	Thumb/Thumb-2	Thumb/Thumb-2 (subconjunto)	Thumb/Thumb-2 (subconjunto)
Pipeline	3 etapas	3 etapas	2 etapas
DSP	Si	No	No
FPU	Si	No	No
MPU	Opcional	No	Opcional

En cuanto a la eficiencia en el manejo del sistema multinúcleo, el LPC54102 también saca ventaja. Gracias a la experiencia obtenida con la serie LPC43xx, no sólo el hardware ha sido mejorado, sino que también las librerías de software brindadas por el fabricante simplifican en gran medida la comunicación entre ambos núcleos.

2.2. Sensor de corriente

Para realizar las pruebas de consumo se utilizó un módulo de bajo costo. En particular, se escogió el sensor de corriente INA219 [12] debido a su facilidad de uso y a que posee una resolución adecuada para realizar las mediciones.

A continuación, se detallan sus principales características:

- Voltaje de alimentación: 3 V – 5 V
- Rango de medición de voltaje: 26 V
- Rango de medición de corriente: $\pm 3,2$ A
- Resolución: 0,8 mA
- Resistencia de sensado: 0,1 ohm 1% 2W
- Protocolo de comunicación: I2C

3. Modelo de Comunicación entre núcleos (IPC)

Tanto en el microcontrolador LPC4337 como en el LPC54102, el Cortex-M4 funciona como procesador maestro, mientras que los Cortex-M0 y Cortex-M0+ funcionan como procesadores esclavos. A su vez, en ambos casos, los dos núcleos tienen acceso completo² al mapa de memoria. Luego de un *reset* del sistema, o al salir del modo de ahorro de energía *Deep power-down*, únicamente el procesador maestro comienza la ejecución de código, mientras que el coprocesador, o procesador esclavo, se mantiene en estado de *reset* hasta que el procesador principal lo despierte. Una vez que ambos procesadores se encuentran ejecutando sus respectivos programas, es posible intercambiar mensajes entre ellos mediante la lectura/escritura de datos en una zona de memoria a la que ambos núcleos tengan acceso. Si bien es posible que cada procesador trabaje por encuesta (polling) de esa zona de memoria para saber si el otro núcleo dejó un mensaje, una mejor aproximación sería utilizar las *interrupciones cruzadas* que cada procesador posee.

En el caso del LPC4337, la posición IRQ#1 del vector de interrupciones del procesador maestro corresponde a la solicitud de interrupción del procesador esclavo, y la posición IRQ#1 del vector de interrupciones del procesador esclavo corresponde a la solicitud de interrupción del procesador maestro. Tanto en el Cortex-M4 como en el Cortex-M0, la solicitud de interrupciones se genera usando la instrucción Assembler *SEV* (Send EVent: Enviar Evento). Dicha instrucción genera un pulso de un clico de reloj en la línea TXEV (Transmit EVent: Transmitir Evento) disponible en cada uno de los núcleos (ver Figura 3.1).

El LPC54102, en cambio, como se mencionó en el apartado 2.1, posee ciertas mejoras que hacen de la comunicación entre núcleos un proceso mucho más sencillo. Cada procesador puede producir hasta 32 interrupciones definidas por el usuario hacia

² A no ser que la Unidad de Protección de Memoria (MPU) de alguno de los procesadores (Cortex-M4 o Cortex-M0+) se haya configurado para limitarle el acceso a alguna región. El Cortex-M0 no dispone de MPU, por lo tanto, no existe forma de limitar su acceso al mapa de memoria del sistema.

el otro núcleo. Para simplificar este proceso, el LPC54102 define (vía software) una estructura de memoria llamada *Mailbox* (Buzón [14]), la cual cuenta con una serie de herramientas para manejar, de una manera simple y con librerías de alto nivel, la comunicación entre el procesador maestro y el esclavo. De esta forma, es posible enviar un mensaje y al mismo tiempo generar una interrupción en la posición IRQ#31 del vector de interrupciones del otro núcleo. A su vez, el LPC54102 posee un registro de exclusión mutua que le permite a los núcleos realizar un acceso seguro a la zona de memoria compartida.

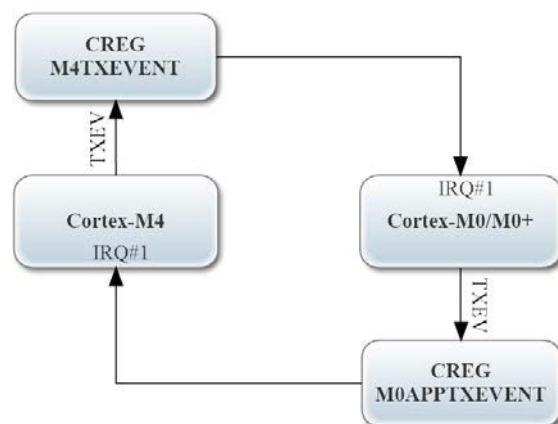


Figura 3.1: Esquema de interrupciones entre núcleos del microcontrolador LPC4337 [13].

De esta manera se evita que los núcleos tengan que encuestar permanentemente una zona de memoria. El procesador que requiera enviar un dato al otro, coloca el mensaje en la memoria compartida y genera un pulso de interrupción. El otro procesador podrá entonces, en la rutina de atención de interrupciones, acceder a la zona de memoria en la que se encuentra el dato y retirarlo.

4. Metodología

4.1. Latencia de comunicación entre núcleos

Para realizar las mediciones de tiempo, se utilizó el registro DWT_CYCCNT perteneciente a la unidad DWT (Data Watchpoint and Trace [15]) implementada en los Cortex-M4F de ambos microcontrolares. Dicho registro cuenta la cantidad de ciclos de ejecución del procesador.

Mediante dicho registro, se obtuvo la cantidad de ciclos que tardó cada prueba en llevarse a cabo y, mediante un cálculo adecuado, se convirtió esa cantidad de ciclos a unidades de tiempo. Este proceso puede observarse en el Listado 3.1.

Listado 3.1: Método para medir latencia entre núcleos.

```

*DWT_CTRL |= 1; // Activa el contador de ciclos
*DWT_CYCCNT = 0; // Resetea el contador de ciclos
/*
TODO: Comunicación entre núcleos
*/
ciclos = (float)(*DWT_CYCCNT); // Latencia en ciclos
TiempoUs = CyclesToUc(ciclos); // Latencia en Us

```

4.2. Consumo de energía

Para realizar las mediciones de consumo energético, se utilizó un método de medición de consumo invasivo, poniendo en serie a la alimentación de las placas de desarrollo el sensor INA219 mencionado en el inciso 2.2. Este sensor devuelve una magnitud analógica proporcional al flujo de corriente que ingresa a la placa que se está queriendo medir. Dicho valor de salida se capturó utilizando el convertidor analógico/digital de otro microcontrolador y que hizo las veces de monitor, mostrando un promedio del consumo durante un tiempo específico.

Cabe aclarar que, para esta primera instancia, no fueron desactivados completamente todos los periféricos de las placas de desarrollo, ni tampoco el procesador de depuración. Por dichas razones, los valores obtenidos no son meramente resultado del consumo de los procesadores.

5. Ensayos y Resultados

5.1. Latencia de comunicación entre núcleos

Con el fin de medir la latencia de comunicación entre núcleos utilizando diferentes métodos de comunicación, se plantearon los siguientes escenarios:

- Escenario A: tiempo que tarda el procesador principal en despertar al coprocesador.
- Escenario B: tiempo de ida y vuelta (RTT) entre núcleos utilizando interrupciones. Para generar las señales de interrupciones se utilizó directamente la instrucción Assembler *SEV* con el fin de reducir al mínimo el tiempo de comunicación. Cabe destacar que en este caso no se realizaron transferencias de datos por lo que no hubo accesos a memoria. Esta prueba sólo fue realizada para el LPC4337, debido a que el manejador de interrupción del LPC54102 no cuenta con ninguna posición que responda ante esta señal.

- Escenario C: este escenario es similar al escenario B, pero en vez de utilizar directamente la instrucción *SEV*, se utilizaron funciones de LPCOpen³ con el fin de evaluar el overhead que agrega utilizar dicha biblioteca.
- Escenario D: en este escenario se midió el tiempo de RTT entre núcleos utilizando la memoria compartida. La comunicación se realizó haciendo polling sobre una dirección de memoria hasta que el otro núcleo escriba un dato.
- Escenario E: este escenario intenta reflejar lo que sería un caso de comunicación real. Se midió el tiempo de RTT entre núcleos utilizando todas las funciones de LPCOpen y, a diferencia del escenario B, sí se realizaron transferencia de datos por lo que este tiempo también refleja varios accesos a memoria.

Tabla 5.1: Tiempos de latencia de las comunicaciones entre núcleos expresados en microsegundos (us).

Escenario	EDU-CIAA-NXP (204MHz)	picoCIAA (96MHz)
A	24,0409	11,0614
B	0,392156	-
C	3,1911	2,0937
D	0,2442	0.2916
E	6,1323	2,7943

La tabla 5.1 muestra que, en la mayoría de los escenarios planteados, con el procesador LPC54102 se obtuvieron mejores tiempos de respuesta, a pesar de estar trabajando a menor velocidad de reloj. Estos resultados responden a lo esperado debido a las mejoras (descriptas en las secciones 2.1 y 3) que posee este procesador frente a su antecesor.

5.2. Consumo de energía

Para medir el consumo energético de ambas placas de desarrollo, también se plantearon diferentes escenarios:

- Escenario A: ambos procesadores se encuentran en reposo.
- Escenario B: procesador maestro realizando procesamiento y procesador esclavo en reposo.
- Escenario C: procesador maestro en reposo y procesador esclavo realizando procesamiento.
- Escenario D: ambos procesadores realizando procesamiento.

³ LPCOpen es una extensa colección de bibliotecas de software (manejadores y middleware) y programas de ejemplo que permiten a los desarrolladores crear productos multifuncionales para la familia de microcontroladores LPC [16].

En las pruebas realizadas sobre la picoCIAA, se estableció la frecuencia de reloj del LPC54102 a su valor máximo, 100 MHz. En lo que respecta a la EDU-CIAA-NXP, las pruebas también se realizaron a su frecuencia de reloj máxima, 204 MHz y, luego, a 100 MHz con el objetivo de comparar los resultados con los obtenidos para la picoCIAA. Con el fin de constatar que los resultados obtenidos sean fehacientes, se tomó como referencia el trabajo realizado por uno de los ingenieros responsables del Proyecto CIAA [17].

Tabla 5.2: Consumo energético de las placas de desarrollo expresado en miliamperios (mA).

Escenario	EDU-CIAA-NXP (204 MHz)	EDU-CIAA-NXP (96 MHz)	picoCIAA (96 MHz)
A	278,16	187,51	20,01
B	299,19	197,71	27,05
C	291,78	194,06	23,9
D	312,25	204,2	31,1

La Tabla 5.2 muestra una amplia diferencia de consumo entre ambas placas de desarrollo. Uno de los motivos de estos resultados es la diferencia de tecnologías que hay entre placas, El LPC54102 tiene mejoras de hardware orientadas a disminuir el consumo. A su vez, otro factor que marca la diferencia es la cantidad de hardware extra que contiene la placa EDU-CIAA-NXP en su conjunto frente a la picoCIAA que, como se mencionó anteriormente, no fueron desactivados en su totalidad para este análisis.

6. Conclusiones y trabajo futuro

6.1. Conclusiones

Este trabajo muestra el funcionamiento de dos microcontroladores con procesadores multinúcleo asimétricos. Se realizaron distintos estudios sobre ambos con el fin de observar cómo respondía cada uno de ellos ante los mismos escenarios de prueba.

Teniendo en cuenta los tiempos de respuesta obtenidos en las pruebas de comunicación entre núcleos, se puede concluir que, dichos tiempos, en ambos microcontroladores, están dentro de los límites correctos y esperados para este tipo de sistemas [18]. Esto da la posibilidad de usar estos microcontroladores en sistemas de tiempo real, en los cuales se necesita tener un alto nivel de determinismo y en donde los tiempos de respuesta son críticos.

En lo que respecta a las pruebas de consumo energético, los resultados obtenidos para la EDU-CIAA-NXP, aunque estén dentro de los valores esperados, no satisfacen los requerimientos de una aplicación de bajo consumo. Sin embargo, las pruebas realizadas sobre la picoCIAA verifican las mejoras tecnológicas del procesador LPC54102, las cuales se ven reflejadas en un menor consumo ante mismos escenarios.

Para finalizar, podemos concluir que la utilización de microcontroladores multinúcleo en sistemas de tiempo real otorga muchas ventajas y amplía significativamente el panorama de aplicaciones a resolver con este tipo de sistemas. Sin embargo, esto conlleva un mayor esfuerzo en el desarrollo. Trabajar sobre este tipo de procesadores implica un nuevo paradigma de programación que no es el habitual en este tipo de sistemas. Para alcanzar un rendimiento adecuado, se deben tener en cuenta todas las condiciones que impone la programación de sistemas paralelos.

6.2. Trabajo futuro

Quedan como pendientes a este trabajo, la medición de consumo al aislar los distintos periféricos y el debugger de cada placa de desarrollo, lo que permitirá hacer foco solamente en el consumo de cada uno de los núcleos. A su vez, será de interés volver a realizar las pruebas utilizando los distintos modos de bajo consumo de ambos procesadores.

Otro de los principales puntos de estudio para el futuro es la utilización de sistemas operativos de tiempo real, estáticos o dinámicos, en microcontroladores multinúcleo. Esto implica el manejo de tareas para cada núcleo, la posibilidad de contar con uno o varios planificadores, sincronización, definición memoria compartida para crear tareas dinámicas, entre otras características importantes para el análisis.

Por último, es de interés el desarrollo de un algoritmo inteligente que pueda decidir en tiempo real qué núcleo debe ejecutar determinada operación, en función a la carga de procesamiento que se requiera. Con esto, se obtendría el mejor rendimiento del procesador, considerando tiempo de respuesta y consumo energético.

Bibliografía

- [1] Fan, X. *Real-Time Embedded Systems: Design Principles and Engineering Practices*. Newnes, 2015. ISBN – 13: 978-0128015070
- [2] Kopetz, H. *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Springer, 2011. ISBN – 13: 978-1441982360
- [3] Laplante, P. *Real-Time Systems Design and Analysis: Tools for the Practitioner*. Wiley-IEEE Press, 2011. ISBN – 13: 978-0470768648
- [4] Raj, P., Raman, A. *The Internet of Things: Enabling Technologies, Platforms, and Use Cases*. Auerbach Publications, 2017. ISBN – 13: 978-1498761284
- [5] Yiu, J. *The Definitive Guide to ARM R Cortex R-M3 and Cortex R-M4 Processors*. Newnes, 2014. ISBN – 13: 978-0124080829
- [6] NXP (2017, Jul 1). *LPC Dual-Core microcontrollers based on ARM® Cortex® -M0 & M4F* [Online]. Disponible en: http://www.nxp.com/pages/lpc-dual-core-cortex-m0-m4f:MC_1423136028688
- [7] NXP (2017, Jul 2). *Product Longevity* [Online]. Disponible en: http://www.nxp.com/about/about-nxp/technology-programs/product-longevity:PRDCT_LONGEVITY_HM
- [8] Proyecto CIAA (2017, Jul 5). *Computadora Industrial Abierta Argentina* [Online]. Disponible en: <http://www.proyecto-ciaa.com.ar>

- [9] Proyecto CIAA (2017, Jul 5). *EDU-CIAA-NXP* [Online]. Disponible en: http://www.proyecto-ciaa.com.ar/index_comprar_educiaanxp.html
- [10] Proyecto CIAA (2017, Jul 5). *picoCIAA* [Online]. Disponible en: http://www.proyecto-ciaa.com.ar/index_comprar_picociaa.html
- [11] ARM (2017, Jul 16). Cortex-M0+ [Online]. Disponible en: <https://developer.arm.com/products/processors/cortex-m/cortex-m0-plus>
- [12] Texas Instruments (2017, Jul 7). *Hoja de datos del sensor de corriente INA219* [Online]. Disponible en: <https://cdn-shop.adafruit.com/datasheets/ina219.pdf>
- [13] *LPC43xx User Manual*, pp. 29, sec. 2.2.
- [14] *LPC5410x User Manual*. Chapter 29.
- [15] ARM (2017, Jul 16). *Cortex-M4 Technical Reference Manual*. Chapter 9. Disponible en: <http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0439b/BABJFFGJ.html>
- [16] NXP (2017, Jul 18). *LPCOpen Libraries and Examples* [Online]. Disponible en: <http://www.nxp.com/products/microcontrollers-and-processors/arm-processors/lpc-cortex-m-mcus/software-tools/lpcopen-libraries-and-examples:LPC-OPEN-LIBRARIE>
- [17] Ridolfi, P. *Extensión del Sistema Operativo FreeOSEK para multiprocesadores asimétricos*. 2015.
- [18] Medina, S., Pi Puig, M., Dell'Oso, M., Romero, F., De Giusti, A., Tinetti, F.: *Comparación de Sistemas Operativos Embebidos sobre la Computadora Industrial Abierta Argentina*. In: Libro de Actas XXI Congreso Argentino de Ciencias de la Computación, pp. 1083 - 1092, 2016. <http://sedici.unlp.edu.ar/handle/10915/56385>